UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/862,612	05/23/2001	Michael J. Lemon	10012649-1	8359

7590 12/23/2005

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER
STEVENS, ROBERT

ART UNIT	PAPER NUMBER
2176	

DATE MAILED: 12/23/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/862,612
Filing Date: May 23, 2001
Appellant(s): LEMON ET AL.

Jonathan Harris
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 8/30/05 appealing from the Office action mailed 5/23/05.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of invention contained in the brief is substantially correct, however the Office provides the following characterization of Appellant's subject matter:

Office Note

Appellant's alleged inventive subject matter describes a mechanism for interfacing among network-based computer entities (e.g., eCommerce sites and customers) that may utilize disparate application layer communication protocols. A centralized conversation controller/manager receives a “message” (e.g., an XML document) from a sending party and determines the protocol (e.g., RosettaNet, Commerce XML, etc.) necessary for the sending party to communicate with a receiving party. Appellant’s alleged inventive subject matter may also track the state of the conversation between the parties. The technical field is related to Electronic Data Interchange (EDI).

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant’s statement of the grounds of rejection to be reviewed on appeal is correct in regard to the claims.

However, Appellant’s brief presents arguments relating to an objection to the drawings. This issue relates to petitionable subject matter under 37 CFR 1.181 and not to appealable subject matter. See MPEP § 1002 and § 1201.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

2002/0161688	Stewart et al.	10-2002

2004/0221292	Chiang et al.	11-04

LeMay, Laura, et al., Sam's Teach Yourself Java 2 in 21 Days, Sam's Publishing, Indianapolis, IN, (c) 1999, pp. 422-430.

(9) Grounds of Rejection

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 4-5, 7-18 and 21-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stewart et al (US Patent Application Publication No. 2002/0161688, relying upon provisional applications filed Feb. 16 and Dec. 29, 2000, hereafter referred to as "Stewart") in view of Chiang et al (US Patent Application Publication No. 2004/0221292, provisionally filed Aug. 8, 2000, hereafter referred to as "Chiang").

Regarding independent claim 1, Stewart discloses:

> *A method for implementing a conversation between a client and a service on a conversation controller, comprising:*

> *the conversation controller receiving a conversation information from the service, the conversion information specifying a structure of conversations supported by the service* (Fig. 23 #430, esp. "Transformation mappings")*;*
>
> *the conversation controller determining a current state of the conversation* ([0144], re: maintaining conversation status)*;*
>
> *the conversation controller using the received conversation information to determine valid input document types for the current state* ([0151] re: "knows how to handle the type of message received")*;*
>
> *the conversation controller verifying whether the message is of one of the valid input document types for the current state* ([0151] re: "knows how to handle the type of message received")*; and*
>
> *the conversation controller dispatching the message to appropriate service entry points provided by the service, until the service produces an output document of a valid output document type.* ([0247] re: "selects a subset of <trading partner> nodes" and [0256] re: "until all filters return true")

However, Stewart does not explicitly disclose:

> *the conversation controller receiving a message on behalf of the service;*

Chiang, though, discloses:

> *the conversation controller receiving a message on behalf of the service;* (Abstract, especially the 2nd sentence)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Chiang for the benefit of Stewart, because to do so would allow a programmer to integrate dissimilar applications, as taught by Chiang in [0010]. These references were all applicable to the same field of endeavor, i.e., the transferring of eCommerce messages among computer platforms.

Regarding claim 4, which is dependent upon claim 1, the limitations of claim 1 have been previously addressed.

However, Stewart does not explicitly disclose:

> *further comprising the conversation controller formatting and returning to the client the output document in a form appropriate to the client.*

Chiang, though, discloses:

> *further comprising the conversation controller formatting and returning to the client the output document in a form appropriate to the client.* (Abstract, re: (ii) converting from server format/source language to end user format/target language)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Chiang for the benefit of Stewart, because to do so would allow a programmer to integrate dissimilar applications, as taught by Chiang in [0010]. These references were all applicable to the same field of endeavor, i.e., the transferring of eCommerce messages among computer platforms.

Regarding claim 5, which is dependent upon claim 1, the limitations of claim 1 have been previously addressed.

Stewart further discloses:

> *the conversation controller calculating a new state of the conversation from the valid output document type;* ([0144] re: maintains conversation status)

However, Stewart does not explicitly disclose:

> *the conversation controller determining new input document types that are valid in the new state; and*
> *the conversation controller prompting for the new input document types that are valid in the new state.*

Chiang, though, discloses:

> *the conversation controller determining new input document types that are valid in the new state;* ([0031] re: invoking type descriptor ... of source and target languages ....) *and*
> *the conversation controller prompting for the new input document types that are valid in the new state.* ([0031] re: invoking type descriptor ... of source and target languages ....)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Chiang for the benefit of Stewart, because to do so would allow a programmer to integrate dissimilar applications, as taught by Chiang in [0010]. These references were all applicable to the same field of endeavor, i.e., the transferring of eCommerce messages among computer platforms.

Regarding claim 7, which is dependent upon claim 1, the limitations of claim 1 have been previously addressed.

Stewart further discloses:

> *further comprising the conversation controller maintaining a "state" of the conversation.* ([0144] re: maintains conversation status)

Regarding claim 8, which is dependent upon claim 1, the limitations of claim 1 have been previously addressed.

Stewart further discloses:

> *further comprising the conversation controller retrieving a "state" of the conversation from the service.* ([0144] re: maintains conversation status)

Regarding claim 9, which is dependent upon claim 1, the limitations of claim 1 have been previously addressed.

Stewart further discloses:

> *the conversation controller calculating a new state of the conversation from the valid output document type;* ([0144] re: maintains conversation status) *and*
> *the conversation controller invoking client methods that can produce new input documents that are valid in the new state.* ([0162] re: business logic plug-ins)

Regarding claim 10, which is dependent upon claim 9, the limitations of claim 9 have been previously addressed.

However, Stewart does not explicitly disclose:

> *further comprising the conversation controller sending the new input documents to the service.*

Chiang, though, discloses:

> *further comprising the conversation controller sending the new input documents to the service.* (Abstract, especially the 2nd sentence)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Chiang for the benefit of Stewart, because to do so would allow a programmer to integrate dissimilar applications, as taught by Chiang in [0010]. These references were all applicable to the same field of endeavor, i.e., the transferring of eCommerce messages among computer platforms.

Regarding independent claim 11, Stewart discloses:

A conversation controller that implements a conversation between a client and a service, comprising:

a processor ([0303] – [0304], esp. "XML Processing")*;*

... , wherein the incoming context handler is capable of parsing the message and extracting a document type of the message ([0109], re: processing protocol specific headers)*;*

an interaction handler executing on said processor ([0303] – [0304], esp. "XML Processing", it being inherent/implicit that a software module executes on a processor) *and coupled to the incoming context handler and capable of identifying a current state* ([0144], re: maintaining conversation status), *and validates the document type based on a ... received from the service* (Fig. 19 #516 and [0270] – [0294] disclose document type/conversation validation)*; and*

a dispatch handler executing on said processor ([0303] – [0304], esp. "XML Processing", it being inherent/implicit that a software module executes on a processor), *wherein the dispatch handler parses* (Fig. 21, re: C-Hub router) *the ... and forwards the message to service entry points of the service* (Fig. 21, re: C-Hub transport).

However, Stewart does not explicitly disclose:

an incoming context handler executing on said processor, said incoming context handler receives a message on behalf of the service, ... ;

... ;

... , conversation specification ...; and

... conversation specification ... .

Chiang, though, discloses:

an incoming context handler executing on said processor, said incoming context handler receives a message on behalf of the service (Abstract, especially the 2nd sentence)*, ... ;*

... ;

... , conversation specification ([0031] re: type descriptor and language metamodels) *...; and*

... conversation specification ([0031] re: type descriptor and language metamodels) *... .*

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Chiang for the benefit of Stewart, because to do so would allow a programmer to integrate dissimilar applications, as taught by Chiang in [0010]. These references were all applicable to the same field of endeavor, i.e., the transferring of eCommerce messages among computer platforms.

Regarding claim 12, which is dependent upon claim 11, the limitations of claim 11 have been previously addressed.

Stewart further discloses:

> *wherein the interaction handler validates if the document type of the message is valid for the current state.* ([0144] re: maintains conversation status)

Regarding claim 13, which is dependent upon claim 11, the limitations of claim 11 have been previously addressed.

Stewart further discloses:

> *wherein the interaction handler calculates a new state of the conversation* ([0144] re: maintains conversation status) *and ... .*

However, Stewart does not explicitly disclose:

> *... and new valid document types for the new state from a response returned by the service.*

Chiang, though, discloses:

> ... *and new valid document types for the new state from a response returned by the service.* ([0031] re: invoking type descriptor ... of source and target languages)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Chiang for the benefit of Stewart, because to do so would allow a programmer to integrate dissimilar applications, as taught by Chiang in [0010]. These references were all applicable to the same field of endeavor, i.e., the transferring of eCommerce messages among computer platforms.

Regarding claim 14, which is dependent upon claim 13, the limitations of claim 13 have been previously addressed.

Stewart further discloses:

> *wherein the interaction handler validates if the document type of the message is valid for the current state.* (Fig. 21 #422 re: XOCP MSGENCODER)

Regarding claim 15, which is dependent upon claim 11, the limitations of claim 11 have been previously addressed.

Stewart further discloses:

> *further comprising a client interaction handler that dispatches a reply from the service to the client and forwards a response from the client to the service.* (Fig. 21 re: "C-Hub Transport")

Regarding independent claim 16, Stewart discloses:

A computer readable medium comprising instructions for implementing a conversation between a client and a service, the instructions comprising:

receiving a ... from the service, the conversation specification specifying a structure of conversations supported by the service (Fig. 23 #430, esp. "Transformation Mappings")*;*

determining a current state of the conversation ([0144], re: maintaining conversation status)*;*

using ..., determining valid input document types for the current state ([0151] re: "knows how to handle the type of message received")*;*

verifying whether the message is of one of the valid input document types for the current state ([0151] re: "knows how to handle the type of message received")*; and*

dispatching the message to appropriate service entry points provided by the service, until the service produces an output document of a valid output document type. ([0247] re; "selects a subset of <trading partner> nodes" and [0256] re: "until all filters return true")

However, Stewart does not explicitly disclose:

... conversation specification ... ;
receiving a message on behalf of the service;
... conversation specification ... ;

Chiang, though, discloses:

... conversation specification ... ; ([0031] re: type descriptor and language metamodels)

receiving a message on behalf of the service; (Abstract, especially the 2nd sentence)

... conversation specification ... ; ([0031] re: type descriptor and language metamodels)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Chiang for the benefit of Stewart, because to do so would allow a programmer to integrate dissimilar applications, as taught by Chiang in [0010]. These references were all applicable to the same field of endeavor, i.e., the transferring of eCommerce messages among computer platforms.

Regarding claim 17, this claim is substantially similar to claim 4, and therefore likewise rejected.

Regarding claim 18, this claim is substantially similar to claim 5, and therefore likewise rejected.

Regarding claim 21, which is dependent upon claim 1, the limitations of claim 1 have been previously addressed.

Stewart further discloses:

> *the conversation controller receiving a ... from the client defining the valid interactions with the client.* (Fig. 7 #196)

However, Stewart does not explicitly disclose:

> *... conversation specification ... .*

Chiang, though, discloses:

> *... conversation specification ... .* ([0031] re: type descriptor and language metamodels)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Chiang for the benefit of Stewart, because to do so would allow a programmer to integrate dissimilar applications, as taught by Chiang in [0010]. These references were all applicable to the same field of endeavor, i.e., the transferring of eCommerce messages among computer platforms.

Regarding claim 22, which is dependent upon claim 16, the limitations of claim 16 have been previously addressed.

Stewart further discloses:

> *wherein the instructions further comprise receiving a ... from the client defining the valid interactions with the client.* (Fig. 7 #196)

However, Stewart does not explicitly disclose:

> *... conversation specification ... .*

Chiang, though, discloses:

> *... conversation specification ... .* ([0031] re: type descriptor and language metamodels)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Chiang for the benefit of Stewart, because to do so would allow a programmer to integrate dissimilar applications, as taught by Chiang in [0010]. These references were all applicable to the same field of endeavor, i.e., the transferring of eCommerce messages among computer platforms.

Claims 2-3 and 19-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stewart et al (US Patent Application Publication No. 2002/0161688, relying upon provisional applications filed Feb. 16 and Dec. 29, 2000, hereafter referred to as "Stewart") in view of Chiang et al (US Patent Application Publication No. 2004/0221292, provisionally filed Aug. 8, 2000, hereafter referred to as "Chiang") and further in view of Laura LeMay et al (Sams Teach Yourself Java 2 in 21 Days, Sams Publishing, Indianapolis, IN, © 1999, pp. 422-430, hereafter referred to as "LeMay")

Regarding claim 2, which is dependent upon claim 1, the limitations of claim 1 have been previously addressed.

Stewart further discloses:

> *wherein if messages of invalid input documents types are received* ([0144] re: errors), ...

However, Stewart does not explicitly disclose:

> ... , *further comprising the conversation controller raising exceptions.*

LeMay, though, discloses:

> ... , *further comprising the conversation controller raising exceptions.* (Throwing exceptions is a well known programming practice. See the p. 426 section entitled "Throwing Exceptions")

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of LeMay for the benefit of Stewart in view of Chiang, because to do so would enable a programmer to handle different types of errors (including custom exceptions), as taught by LeMay in the first paragraph under section "Creating Your Own Exceptions" on p. 427. These references were all applicable to the same field of endeavor, i.e., object oriented programming.

Regarding claim 3, which is dependent upon claim 1, the limitations of claim 1 have been previously addressed.

Stewart further discloses:

> *wherein if no valid output document is produced by the service* ([0144] re: errors), ...

However, Stewart does not explicitly disclose:

> ... , *further comprising the conversation controller raising exceptions.*

LeMay, though, discloses:

> ... , *further comprising the conversation controller raising exceptions.* (Throwing exceptions is a well known programming practice. See the p. 426 section entitled "Throwing Exceptions")

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of LeMay for the benefit of Stewart in view of Chiang, because to do so would enable a programmer to handle different types of errors (including custom exceptions), as taught by LeMay in the first paragraph under section "Creating Your Own Exceptions" on p. 427. These references were all applicable to the same field of endeavor, i.e., object oriented programming.

Regarding claims 19-20, these claims are substantially similar to claims 2-3, respectively, and therefore likewise rejected.

(10) Response to Argument

Commencing on page 10 of the Appeal Brief (hereafter "the Brief"), Appellant presents the following issues, which are accordingly addressed below.

A. Claims 1, 4-5, 7-10 and 21

On pages 10-12, the Appellant first sets out a complaint as to a typographical error in the Final rejection.

The Office apologized for and orally corrected said typographical error in the telephonic interview held with Appellant's Attorney on August 1, 2005. The Office also included a verbatim relevant passage from the correct paragraph of the reference in the written rejection (which enabled Appellant to locate the intended passage even before the Feb. 7, 2005 filing of an Amendment after Non-Final Rejection). It is noted that the Appellant did, in fact, locate the intended passage, thus rendering this issue moot. Additionally, the correct paragraph is now explicitly referenced in this communication. The Office does not consider these comments to be germane to the issues on appeal.

The Appellant next argues that there is no teaching in the cited references of "a current state" or "verifying whether the message is one of the valid input document types for the current state".

The Office respectfully disagrees. First, the cited statement indicating that the subject matter of the Stewart reference "knows how to handle the message being received" discloses that the message is in a "current state" (i.e., "being received") and that the message is “verified” (i.e., “handled” or sent to the appropriate handler). It is inherent, or at least implied, that Stewart must determine a current state of a message and validate the message type, else Stewart, which provides network-based collaboration among business systems, would not function. Further, the Office respectfully notes the following figure elements depicted in Stewart: Fig. 6 #190 (showing a “Conversation Management” element), Figure 8 #126 and 218 (showing elements hosting Conversation Management/Business Management functions and making use of the standard XOCP messaging protocol), Fig. 14 (business protocol states), and paragraph [0324] which provides a DTD excerpt showing the use of state and message type. The Office also notes the similarity of Appellant’s Figure 5 and the Stewart Figure 3.

The Appellant further argues that there is no teaching in the cited references of “the conversation controller receiving a conversation information from the service, the conversation information specifying a structure of conversations supported by the service”.

The Office respectfully disagrees. First, the cited statement indicating that the subject matter of the Stewart reference “knows how to handle the message being received” discloses that the message has been received and that the appropriate conversation protocol is chosen to implement communications between communicating entities. It is inherent, or at least implied, that Stewart must determine the appropriate conversation protocol, else Stewart, which provides network-based collaboration among business systems, would not function. Further, the Office respectfully notes the following figure elements depicted in Stewart: Fig. 6 #190 (showing a "Conversation Management" element), Figure 8 #126 and 218 (showing elements hosting Conversation Management/Business Management functions and making use of the standard XOCP messaging protocol), Fig. 4 #144 (selecting one or more XML Vocabularies for Inter-Business Collaboration), and paragraph [0324] which provides a DTD excerpt showing the use of state and message type. The Office also notes the similarity of Appellant's Figure 5 and the Stewart Figure 3.

The Appellant next sets out an argument regarding the withdrawal of rejections under 35 USC §101.

Since the Office withdrew the rejections raised under 35 USC §101, the Office considers these arguments to be moot. The Office does not consider these comments to be germane to the issues on appeal.

B. Claims 11-15

On page 12, the Appellant argues that there is no teaching in the cited references of an element that "validates the document type based on a conversation specification received from the service".

The Office respectfully disagrees. First, the cited statement indicating that the subject matter of the Stewart reference "knows how to handle the message being received" discloses that the message is "validated" (i.e., "handled" or sent to the appropriate handler). It is inherent, or at least implied, that Stewart must determine and validate the message type, else Stewart, which provides network-based collaboration among business systems, would not function. Further, the Office respectfully notes the following figure elements depicted in Stewart: Fig. 6 #190 (showing a "Conversation Management" element), Figure 8 #126 and 218 (showing elements hosting Conversation Management/Business Management functions and making use of the standard XOCP messaging protocol), Fig. 14 (business protocol states), and paragraph [0324] which provides a DTD excerpt showing the use of state and message type. The Office also notes the similarity of Appellant's Figure 5 and the Stewart Figure 3.

C. Claims 16-18 and 22

On pages 12-13, the Appellant argues that there is no teaching in the cited references of "receiving a conversation specification from the service, the conversation specification specifying a structure of conversations supported by the service" and "using the conversation specification, determining valid input document types for the current state".

On page 12, the Appellant argues that there is no teaching in the cited references of "receiving a conversation specification from the service, the conversation specification specifying a structure of conversations supported by the service" and "using the conversation specification, determining valid input document types for the current state".

The Office respectfully disagrees. Regarding the first limitation, the cited statement indicating that the subject matter of the Stewart reference "knows how to handle the message being received" discloses that the message has been received and that the appropriate conversation protocol is chosen to implement communications between communicating entities. It is inherent, or at least implied, that Stewart must determine the appropriate conversation protocol, else Stewart, which provides network-based collaboration among business systems, would not function. Further, the Office respectfully notes the following figure elements depicted in Stewart: Fig. 6 #190 (showing a "Conversation Management" element), Figure 8 #126 and 218 (showing elements hosting Conversation Management/Business Management functions and making use of the standard XOCP messaging protocol), Fig. 4 #144 (selecting one or more XML Vocabularies for Inter-Business Collaboration), and paragraph [0324] which provides a DTD excerpt showing the use of state and message type

Regarding the second limitation, the cited statement indicating that the subject matter of the Stewart reference "knows how to handle the message being received" discloses that the message is "validated" (i.e., "handled" or sent to the appropriate handler). It is inherent, or at least implied, that Stewart must determine and validate the message type, else Stewart, which provides network-based collaboration among business systems, would not function. Further, the Office respectfully notes the following figure elements depicted in Stewart: Fig. 6 #190 (showing a "Conversation

Management" element), Figure 8 #126 and 218 (showing elements hosting Conversation Management/Business Management functions and making use of the standard XOCP messaging protocol), Fig. 14 (business protocol states), and paragraph [0324] which provides a DTD excerpt showing the use of state and message type.

The Office also notes the similarity of Appellant's Figure 5 and the Stewart Figure 3.

D. Claims 2-3 and 19-20

On page 13, the Appellant asserts the arguments previously raised for independent claims 1 and 16.

The Office has addressed these arguments above.

For at least these reasons, the Office maintains the rejections of the claims under 35 USC §103(a), as set forth above.

E. Figure 1

On page 13, the Appellant asserts that the objection to Figure 1 is incorrect.

However, the Office notes this issue relates to petitionable subject matter under 37 CFR 1.181 and not to appealable subject matter. See MPEP § 1002 and § 1201. Therefore, the objection to Figure 1 is not addressed at this time.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Robert Stevens

December 5, 2005

William L. Bashore
WILLIAM BASHORE
PRIMARY EXAMINER
12/19/2005

Conferees:

Heather Herndon
HEATHER R. HERNDON
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Heather Herndon

Stephen Hong